

# CONVERGENCE OF TOMLIN'S HOTS ALGORITHM

OLIVIER FERCOQ \*

**Abstract.** The HOTS algorithm uses the hyperlink structure of the web to compute a vector of scores with which one can rank web pages. The HOTS vector is the vector of the exponentials of the dual variables of an optimal flow problem (the “temperature” of each page). The flow represents an optimal distribution of web surfers on the web graph in the sense of entropy maximization.

In this paper, we prove the convergence of Tomlin’s HOTS algorithm. We first study a simplified version of the algorithm, which is a fixed point scaling algorithm designed to solve the matrix balancing problem for nonnegative irreducible matrices. The proof of convergence is general (non-linear Perron-Frobenius theory) and applies to a family of deformations of HOTS. Then, we address the effective HOTS algorithm, designed by Tomlin for the ranking of web pages. The model is a network entropy maximization problem generalizing matrix balancing. We show that, under mild assumptions, the HOTS algorithm converges with a linear convergence rate. The proof relies on a uniqueness property of the fixed point and on the existence of a Lyapunov function.

We also show that the coordinate descent algorithm can be used to find the ideal and effective HOTS vectors and we compare HOTS and coordinate descent on fragments of the web graph. Our numerical experiments suggest that the convergence rate of the HOTS algorithm may deteriorate when the size of the input increases. We thus give a normalized version of HOTS with an experimentally better convergence rate.

**1. Introduction.** Internet search engines use a variety of algorithms to sort web pages based on their text content or on the hyperlink structure of the web. In this paper, we focus on an algorithm proposed by Tomlin in [31] for the ranking of web pages, called HOTS. It may also be used for other purposes like the ranking of sport teams [12]. Like PageRank [4], HITS [17] and SALSA [22], HOTS uses the hyperlink structure of the web (see also [20, 21] for surveys on link-based ranking algorithms). This structure is summarized in the web graph, which is a digraph with a node for each web page and an arc between pages  $i$  and  $j$  if there is a hyperlink from page  $i$  to page  $j$ .

The HOTS vector, used to rank web pages, is the vector of the exponentials of the dual variables of an optimal flow problem. The flow represents an optimal distribution of web surfers on the web graph in the sense of entropy maximization. The dual variable, one by page, is interpreted as the “temperature” of the page, the hotter a page the better. In the case of the PageRank, the flow of websurfers is determined by the uniform transition probability of following one hyperlink in the current page. This transition rule is in fact arbitrary. The HOTS model assumes that the web surfers choose the hyperlink to follow by maximizing the entropy of the flow. Tomlin showed that this vector is solution of a nonlinear fixed point equation. He then proposed a scaling algorithm to compute the HOTS vector, based on this fixed point equation.

This algorithm solves the matrix balancing problem studied among others in [13, 8, 29, 28]. Given a  $n \times n$  nonnegative matrix  $A$ , the matrix balancing problem consists in finding a matrix  $X$  of the form  $X = D^{-1}AD$  with  $D$  diagonal definite positive and such that  $\sum_k X_{i,k} = \sum_j X_{j,i}$  for all  $i$ . We shall compare Tomlin’s HOTS algorithm with Schneider and Zenios’s coordinate descent DSS algorithm [29]. The main difference between these algorithms is that in coordinate descent, the scaling is done node by node in the network (i.e. in a Gauss-Seidel fashion) whereas in Tomlin’s HOTS algorithm, the scaling is done all the nodes at the same time, in a Jacobi fashion.

---

\*INRIA Saclay and CMAP Ecole Polytechnique, ([olivier.fercoq@inria.fr](mailto:olivier.fercoq@inria.fr)). The doctoral work of the author is supported by Orange Labs through the research contract CRE 3795 with INRIA

A problem close to the matrix balancing problem is the equivalence scaling problem, where given a  $m \times n$  nonnegative matrix  $A$ , we search for a matrix  $X$  of the form  $X = D_1 A D_2$  with  $D_1$  and  $D_2$  diagonal definite positive and such that  $X$  is bistochastic. The Sinkhorn-Knopp [19] algorithm is a famous algorithm designed for the resolution of the scaling problem. We may see HOTS algorithm as the analog of Sinkhorn-Knopp algorithm for the matrix balancing problem: both algorithms correspond to fixed point iterations on the diagonal scalings. Moreover, Smith [30] and Knight [18] proposed to rank web pages according to the inverse of the corresponding entry in the diagonal scaling.

However, whereas Sinkhorn-Knopp algorithm [19] and the coordinate descent algorithm [23] have been proved to converge, it does not seem that a theoretical result on the convergence of Tomlin's HOTS algorithm has been stated in previous works, although experimentations [31] suggest that it is the case. Indeed, Knight [18, Sec. 5] rose the fact that Tomlin did not state any convergence result for HOTS algorithm. Another algorithm for the matrix balancing problem is given in [14], based on the equivalence between the matrix balancing problem and the problem of minimizing the dominant eigenvalue of an essentially nonnegative matrix under trace-preserving diagonal perturbations [15].

In this paper, we prove the convergence of Tomlin's HOTS algorithm. We first study a simplified version of the algorithm that we call the ideal HOTS algorithm. It is a fixed point scaling algorithm that solves the matrix balancing problem for non-negative irreducible matrices. We prove its convergence thanks to nonlinear Perron-Frobenius theory (Theorem 3.6). The proof methods are general and apply to a family of deformations of HOTS. Then, we address the effective HOTS algorithm, for the general case, which is the version designed by Tomlin for the ranking of web pages. Indeed the web graph is not strongly connected, which implies that the balanced matrix does not necessarily exist. The model is a nonlinear network entropy maximization problem which generalizes matrix balancing. We show in Theorem 4.6 that under mild assumptions the HOTS algorithm converges with a linear rate of convergence. The proof relies on the properties of the ideal HOTS algorithm: uniqueness of the fixed point up to an additive constant and decrease of a Lyapunov function at every step (Theorem 3.9).

We also show that Schneider and Zenios's coordinate descent algorithm can be adapted to find the ideal and effective HOTS vectors. We compare the HOTS algorithm and coordinate descent on fragments of the web graph in Section 6. We considered small, medium and large size problems. In all cases the respective computational costs of both algorithms were similar. As the performances of the HOTS algorithm depends on the primitivity of the adjacency matrix considered and coordinate descent does not, coordinate descent can be thought to have a wider range of applications. However, the actual implementation of the HOTS algorithm is attractive for web scale problems: whereas coordinate descent DSS uses at each iteration (corresponding to a given web page) information from incoming and outgoing hyperlinks, the HOTS algorithm reduces to elementwise operations and left and right matrix vector products. Hence, an iteration of the HOTS algorithm can be performed without computing neither storing the transpose of the adjacency matrix.

We give an exact coordinate descent algorithm for the truncated scaling problem defined in [27] and we extend its use to the problem of computing the HOTS vector when some bounds on the web surfers flow are known. Experimental results show that exact coordinate descent is an efficient algorithm for web scale problems and that it

is faster than the inexact coordinate descent algorithm presented in [28]. Finally, we remarked that the convergence rate of the effective HOTS algorithm seems to deteriorate when the size of the graph considered increases. In order to overcome this feature, we propose a normalized version of the HOTS algorithm where we maximize a relative entropy of the flow of web surfers instead of the classical entropy. A byproduct is that the associated ranking favors pages with no outlink less than Tomlin's HOTS.

The paper is organized as follows. In Section 2, we recall the main theorems of nonlinear Perron-Frobenius theory, in Section 3, we prove the convergence of the ideal HOTS algorithm and we give a Lyapunov function for this algorithms. In Section 4, we give the convergence rate of the effective HOTS algorithm. In Section 5, we study the HOTS problem with bounds on the flow of web surfers. In Section 6, we compare various candidate algorithms to compute the HOTS vector and in Section 7, we give the normalized HOTS algorithm.

**2. Nonlinear Perron-Frobenius theory.** The classical Perron-Frobenius theorem (see [2] for instance) states that the spectral radius of a nonnegative matrix  $A$  is an eigenvalue (called the Perron root) and that there exists an associated eigenvector with nonnegative coordinates. If, in addition,  $A$  is irreducible, then the Perron root is simple and the (unique up to a multiplicative constant) nonnegative eigenvector, called the Perron vector, has only positive entries. The nonlinear Perron-Frobenius theory is an extension of the Perron-Frobenius theorem to monotone and homogeneous maps. It has a multiplicative and an additive formulation.

**DEFINITION 2.1.** *A map  $T : \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$  is monotone if for all vectors  $p, q$  such that  $p \leq q$ ,  $T(p) \leq T(q)$ . A map  $T : \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$  is homogeneous if for all vector  $p$  and for all nonnegative real  $\lambda$ ,  $T(\lambda p) = \lambda T(p)$ .*

**DEFINITION 2.2.** *A map  $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is additively homogeneous if for all vector  $p$  and for all real  $\lambda$ ,  $T(\lambda + p) = \lambda + T(p)$ .*

We can transform a multiplicative monotone, homogeneous map  $T^\times$  into a monotone, additively homogeneous map  $T^+$  and vice versa by the following operation called the “logarithmic glasses”:

$$T^+(p) = \log(T^\times(\exp(p)))$$

where  $\log$  and  $\exp$  act elementwise.

The following results show that monotone and nonexpansive maps are indeed nonexpansive. Hence, they are well suited for iterative algorithms.

**PROPOSITION 2.3 ([7]).** *An additively homogeneous map is nonexpansive for the sup-norm if and only if it is monotone.*

For a more general result, we shall need Hilbert's projective metric.

**DEFINITION 2.4.** *For  $x, y$  two vectors of  $\mathbb{R}^n$ , Hilbert's projective metric between  $x$  and  $y$  is defined as*

$$d(x, y) = \log\left(\max_{i, j \in [n]} \frac{x_i y_j}{y_i x_j}\right)$$

**PROPOSITION 2.5 ([5]).** *Any monotone and homogeneous map is nonexpansive for Hilbert's metric.*

**DEFINITION 2.6.** *For a map  $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$  or  $T : \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ , we call the graph of  $T$  and we denote it  $G(T)$ , the directed graph with nodes  $1, \dots, n$  and an arc from  $i$  to  $j$  if and only if  $\lim_{t \rightarrow +\infty} T(te_i) = +\infty$  where  $e_i$  is the  $i$ th basis vector.*

The following results give conditions for the existence and uniqueness of the “eigenvector” of a monotone, (additively or multiplicatively) homogeneous map.

**THEOREM 2.7** (Theorem 2 in [11]). *Let  $T$  be a monotone, additively homogeneous map. If  $G(T)$  is strongly connected, then there exists  $u \in \mathbb{R}^n$  and  $\lambda \in \mathbb{R}$  such that  $T(u) = \lambda + u$ . We say that  $u$  is an additive eigenvector of  $f$ .*

**THEOREM 2.8** (Corollary 2.5 in [24], Theorem 2.3 in [10], Theorem 6.8 in [1]). *Let  $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$  be a continuously differentiable map which is monotone and additively homogeneous and has an additive eigenvector  $p \in \mathbb{R}^n$ . If  $\nabla T(p)$  is irreducible, then the eigenvector is unique up to an additive factor. If  $\nabla T(p)$  is primitive, then all the orbits defined by*

$$p_{k+1} = T(p_k) - \psi(T(p_k))$$

*for a given additively homogeneous function  $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$  converge to  $p - \psi(p)$  linearly at a rate equal to  $|\lambda_2(\nabla T(p))| = \max\{|\lambda|; \lambda \in \text{spectrum}(\nabla T(p)), \lambda \neq 1\}$ .*

These theorems have been stated with more general assumptions, among others semi-differentiability [1] and infinite state space [24]. However, for the sake of simplicity, we present them here in this simpler form. We can also write them theorem in the multiplicative form.

These results give a general framework to prove that the HOTS score and more generally many web rankings are well defined, i.e. that the score is unique and that the fixed point algorithm (or power algorithm) used to compute them indeed converges to the expected ranking.

**3. The ideal HOTS algorithm.** The web graph is a graph constructed from the hyperlink structure of the web. Each web page is represented by a node and there is an arc between nodes  $i$  and  $j$  if and only if page  $i$  points to page  $j$ . We shall denote by  $A$  the adjacency matrix of the web graph.

There are two versions of the HOTS algorithm: an ideal version for strongly connected graphs, i.e. for irreducible adjacency matrices, and an effective version for general graphs that we will study in Section 4. The HOTS algorithm for irreducible matrices is designed for the resolution of the following nonlinear network flow problem. The optimization variable  $\rho_{i,j}$  represents the traffic of websurfers on the hyperlink from page  $i$  to page  $j$ .

$$\begin{aligned} \max_{\rho \geq 0} & - \sum_{i,j \in [n]} \rho_{i,j} (\log(\frac{\rho_{i,j}}{A_{i,j}}) - 1) \\ & \sum_{j \in [n]} \rho_{i,j} = \sum_{j \in [n]} \rho_{j,i}, \quad \forall i \in [n] & (p_i) \\ & \sum_{i,j \in [n]} \rho_{ij} = 1 & (\mu) \end{aligned}$$

The dual problem consists in minimizing the function  $\theta$  on  $\mathbb{R}^n \times \mathbb{R}$  where

$$\theta(p, \mu) := \sum_{i,j \in [n]} A_{ij} e^{p_i - p_j + \mu} - \mu.$$

We use the convention that  $0 \log(0) = 0$  and that  $x \log(x/0) = 0$  if  $x = 0$  and  $x \log(x/0) = +\infty$  otherwise.

If  $(p, \mu)$  is a minimizer of  $\theta$ , then the value of  $\exp(p_i)$  is interpreted as the temperature of page  $i$ , the hotter the better. We call it the HOTS (Hyperlinked Object Temperature Scale) score.

The ideal HOTS algorithm (Algorithm 1) reduces to the fixed point iterations for the function  $f$  defined by

$$f(x) = \frac{1}{2}(\log(A^T e^x) - \log(Ae^{-x})) . \quad (3.1)$$

Denoting  $y_i = e^{p_i}$ , we can write it in multiplicative form to spare computing the exponentials and logarithms.

---

**Algorithm 1** Ideal HOTS algorithm [31]

---

Start with an initial point  $y_0 \in \mathbb{R}^n$ ,  $y_0 > 0$ . Given  $y^k$ , compute  $y^{k+1}$  such that

$$y_i^{k+1} = \left( \frac{\sum_{j \in [n]} A_{j,i} y_j^k}{\sum_{l \in [n]} A_{i,l} (y_l^k)^{-1}} \right)^{\frac{1}{2}} .$$


---

---

**Algorithm 2** Coordinate descent DSS [29]

---

Start with an initial point  $y^0 \in \mathbb{R}^n$ ,  $y_0 > 0$ . Given  $y^k$ , select a coordinate  $i \in [n]$  and compute  $y^{k+1}$  such that

$$y_i^{k+1} = \left( \frac{\sum_{j \in [n]} A_{j,i} y_j^k}{\sum_{l \in [n]} A_{i,l} (y_l^k)^{-1}} \right)^{\frac{1}{2}} \\ y_j^{k+1} = y_j^k, \quad \forall j \neq i$$


---

We shall compare the HOTS algorithm with Schneider and Zenios's coordinate descent DSS algorithm (Algorithm 2). This is indeed a coordinate descent algorithm since for every  $k$ , we have, denoting  $p_i = \log(y_i)$ ,

$$p_i^{k+1} = \arg \min_{x \in \mathbb{R}} \theta(p_1^k, \dots, p_{i-1}^k, x, p_{i+1}^k, \dots, p_n^k) .$$

Coordinate descent algorithms (Algorithm 3) are designed to solve

$$\min_{x \in \mathcal{X}} \phi(x) \quad (3.2)$$

where  $\mathcal{X}$  is a possibly unbounded box of  $\mathbb{R}^n$  and  $\phi$  has the form  $\phi(x) = \psi(Ex) + \langle b, x \rangle$ ,  $\psi$  is a proper closed convex function,  $E$  is a  $m \times n$  matrix having no zero row and  $b$  is a vector of  $\mathbb{R}^n$ .

---

**Algorithm 3** Coordinate descent

---

Start with an initial point  $x^0 \in \mathbb{R}^n$ . Given  $x^k$ , select a coordinate  $i \in [n]$  and compute  $x^{k+1}$  such that

$$x_i^{k+1} = \arg \min_{l_i \leq y \leq u_i} \phi(x_1^k, \dots, x_{i-1}^k, y, x_{i+1}^k, \dots, x_n^k) \\ x_j^{k+1} = x_j^k, \quad \forall j \neq i$$


---

PROPOSITION 3.1 ([23]). Assume that the set of optimal solutions  $\mathcal{X}^*$  of (3.2) is nonempty, that the domain of  $\psi$  is open, that  $\psi$  is twice continuously differentiable on its domain and that  $\nabla^2\psi(Ex)$  is positive definite for all  $x \in \mathcal{X}^*$ . Let  $(x^k)_k$  be a sequence generated by the coordinate descent algorithm (Algorithm 3), using the cyclic rule (more general rules are also possible). Then  $(x^k)_k$  converges at least linearly to an element of  $\mathcal{X}^*$ .

We now study the fixed point operator  $f$  defined in (3.1).

PROPOSITION 3.2 ([31]). A vector  $p \in \mathbb{R}^n$  is a fixed point of  $f$  defined in (3.1) if and only if the couple  $(p, \mu)$  with  $\mu = -\log(\sum_{i,j \in [n]} A_{ij} e^{p_i - p_j})$  is a minimizer of the dual function. Moreover, in this case, denoting  $D = \text{diag}(\exp(p))$ ,  $e^\mu D A D^{-1}$  is a maximizer of the network flow problem.

*Proof.* As  $\theta$  is convex and differentiable, a couple  $(p, \mu)$  is a minimizer if and only if it cancels the gradient.  $\frac{\partial \theta}{\partial \mu}(p, \mu) = \sum_{i,j \in [n]} A_{ij} e^{p_i - p_j} e^\mu - 1$ , so we have the expression of the optimal  $\mu$  as a function of  $p$ . To conclude, we remark that

$$\frac{\partial \theta}{\partial p_k}(p, \mu) = \left( - \sum_{i \in [n]} A_{i,k} e^{p_i - p_k} + \sum_{j \in [n]} A_{k,j} e^{p_k - p_j} \right) e^\mu = 0$$

is equivalent to  $f(p) = p$ . To get back to the primal problem, we remark that the primal cost of  $e^\mu D A D^{-1}$  is equal to the dual cost of  $(p, \mu)$  and that it is an admissible circulation.  $\square$

PROPOSITION 3.3. The map  $f$  defined in (3.1) is monotone, additively homogeneous (Definition 2.2).

*Proof.* For all real  $\lambda$  and for all vectors  $p, q$  such that  $p \leq q$ ,  $f(\lambda + p) = \lambda + f(p)$  ( $\log(e^\lambda) = \lambda$ ) and  $f(p) \leq f(q)$  ( $\log$  and  $\exp$  are increasing functions).  $\square$

The following result gives the conditions for the existence and uniqueness of the ideal HOTS vector.

THEOREM 3.4 ([8]). There exists  $v \in \mathbb{R}^n$  such that  $f(v) = v$  and  $\sum_{i \in [n]} v_i = 0$  if and only if  $A$  has a diagonal similarity scaling if and only if  $A$  is completely reducible.

If in addition  $A$  is irreducible, then this vector is unique.

COROLLARY 3.5 ([29]). If  $A$  is completely reducible, coordinate descent DSS (Algorithm 2) converges linearly to a vector  $v$  such that  $\text{diag}(v) A \text{diag}(v)^{-1}$  is scaled.

To prove the convergence of the ideal HOTS algorithm (Algorithm 1), we use the nonlinear Perron-Frobenius theory, the main theorems of which are stated in Section 2.

THEOREM 3.6. Let  $f$  be the map defined in (3.1). If  $A$  is irreducible and  $A + A^T$  is primitive, then there exists a vector  $v$  and such that  $f(v) = v$  and for all  $x \in \mathbb{R}^n$ ,

$$\limsup_{k \rightarrow \infty} \|f^{k+1}(x) - v\|^{1/k} \leq |\lambda_2(P)| = \max\{|\lambda|; \lambda \in \text{spectrum}(P), \lambda \neq 1\}$$

where  $P = \frac{1}{2} (\text{diag}(A^T e^v)^{-1} A^T \text{diag}(e^v) + \text{diag}(A e^{-v})^{-1} A \text{diag}(e^{-v}))$ . In particular, the ideal HOTS algorithm (Algorithm 1) converges linearly at rate  $|\lambda_2(P)|$ .

*Proof.* The iterates of the fixed point iteration defined by  $p^0 = x$  and  $p^{k+1} = f(p^k)$  verify  $p^k = \log(y^k)$  where  $y^k$  is the  $k$ th iterate of the ideal HOTS algorithm (Algorithm 1) started with  $y^0 = \exp(x)$ . Hence, by continuous differentiability of the exponential, the rate of convergence of both versions of the algorithm is the same. By Theorem 3.4, as  $A$  is irreducible,  $f$  has a fixed point  $v$  and  $\text{diag}(\exp(v))$  is solution of the matrix balancing problem associated to  $A$ . Now easy calculations show that  $\nabla f = P$ . As  $P$  has the same pattern as  $A + A^T$ ,  $P$  is primitive if and only if  $A + A^T$  is. The result follows from Theorem 2.8.  $\square$

This theorem shows that the HOTS vector for the irreducible case is well defined if  $A$  is irreducible and that if  $A + A^T$  is primitive, then the ideal HOTS algorithm (3.1) converges linearly to the HOTS vector.

REMARK 1. *The ideal HOTS algorithm (Algorithm 1) requires a primitivity assumption in order to converge that coordinate descent DSS (Algorithm 2) does not require. On the other hand, the convergence rate of coordinate descent DSS is not explicitly given while Theorem 3.6 gives the convergence rate of ideal HOTS.*

REMARK 2. *Changing the diagonal of  $A$  does not change the optimal scaling, so we can choose a nonzero diagonal for  $A$  in the preceding theorem. This is useful when  $A$  is irreducible but not primitive.*

The fixed point equation defining the ideal HOTS vector is

$$y_i = \left( \frac{\sum_j A_{j,i} y_j}{\sum_k A_{i,k} y_k^{-1}} \right)^{\frac{1}{2}}.$$

Indeed, the page  $i$  has a good HOTS score if it is linked to by pages with a good HOTS score and if it does not link to pages with a bad HOTS score.

We thus introduce the following set of fixed point ranking algorithms.

---

**Algorithm 4** Deformed HOTS algorithm

---

Let  $\alpha, \beta \geq 0$  such that  $\alpha + \beta = 1$  and let  $g : \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$  defined for all  $i$  by

$$g_i(x) = \frac{(\sum_j A_{j,i} x_j)^\alpha}{(\sum_k A_{i,k} x_k^{-1})^\beta}.$$

Given an initial point  $d_0 \in \mathbb{R}^n$  and a norm  $\|\cdot\|$ , the deformed HOTS algorithm is defined by

$$d^{k+1} = \frac{g(d^k)}{\|g(d^k)\|}$$


---

PROPOSITION 3.7. *Let  $\alpha, \beta \geq 0$  such that  $\alpha + \beta = 1$ . If  $A$  is irreducible and  $\alpha A + \beta A^T$  is primitive, then the deformed HOTS algorithm (Algorithm 4) converges linearly to a positive vector.*

*Proof.* Let  $h = \log \circ g \circ \exp$ . As in the proof of Theorem 4.6, the rate of convergence for the fixed point iterations with  $g$  or  $h$  is the same. The map  $h$  is monotone and additively homogeneous. For  $\alpha > 0$ , its graph is equal to  $A$ . Hence, for  $\alpha > 0$ ,  $h$  has an eigenvector by Theorem 2.7. For  $\alpha = 0$ , as  $A$  is irreducible, by the Perron-Frobenius theorem [2],  $A$  has an eigenvector  $x$ . Then  $\log(x^{-1})$  is an eigenvector of  $h$ . Now,  $\nabla h(v) = \alpha \text{diag}(A^T e^v)^{-1} A^T \text{diag}(e^v) + \beta \text{diag}(A e^{-v})^{-1} A \text{diag}(e^{-v})$ , so we have the convergence as soon as  $\alpha A + \beta A^T$  is primitive by Theorem 2.8.  $\square$

REMARK 3. *For  $\alpha = \frac{1}{2}$ , we have the fixed point diagonal similarity scaling, for  $\alpha = 1$ , we have the ranking by the Perron vector [16] and for  $\alpha = 0$ , we have an “anti-Perron” score, where good pages are those that do not link to pages with a bad score.*

The following result gives a global contraction factor in the case when  $A$  is positive.

PROPOSITION 3.8. *If  $k(A)$  is the contraction factor of  $A$  in Hilbert metric ( $k(A) < 1$  if  $A$  is positive), then  $f$  is  $\frac{k(A^T) + k(A)}{2}$ -contracting in Hilbert metric.*

*Proof.* Let  $x$  and  $y$  be two positive vectors such that  $\eta y \leq x \leq \nu y$  elementwise. Then  $\eta' A^T y \leq A^T x \leq \nu' A^T y$  with  $\log(\nu'/\eta') \leq k(A^T) \log(\nu/\eta)$ . We also have that  $\eta'' A y^{-1} \leq A x^{-1} \leq \nu'' A y^{-1}$  with  $\log((\eta'')^{-1}/(\nu'')^{-1}) \leq k(A) \log(\nu/\eta)$ . Hence,

$$d(g(x), g(y)) = \log\left(\left(\frac{\nu''\nu'}{\eta''\eta'}\right)^{1/2}\right) \leq \frac{k(A^T) + k(A)}{2} \log\left(\frac{\nu}{\eta}\right) . \quad \square$$

A key technical ingredient of the convergence of the effective HOTS algorithm described in the next section will be Theorem 3.9 below showing that each iteration  $p \leftarrow f(p)$  of the ideal HOTS algorithm does not increase the dual objective function.

**THEOREM 3.9** (Lyapunov function).  $\theta(f(p)) \leq \theta(p)$

*Proof.* Let us denote  $\psi(p, q) = \sum_{i,j} e^{p_i} A_{ij} e^{-q_j}$ .

$$\begin{aligned} \psi(p, 2f(p) - p) &= \sum_{i,j} e^{p_i} A_{ij} \left( \frac{(Ae^{-p})_j}{(A^T e^p)_j} \right) e^{p_j} = \sum_j (Ae^{-p})_j e^{p_j} \\ &= \theta(p) = \psi(2f(p) - p, p) \end{aligned}$$

Now, as  $\psi$  is convex,

$$\theta(f(p)) = \psi\left(\frac{1}{2}(2f(p) - p, p) + \frac{1}{2}(p, 2f(p) - p)\right) \leq \theta(p) . \quad \square$$

**4. The effective HOTS algorithm.** Theorem 3.4 gives conditions for the existence and uniqueness of the HOTS vector in the ideal case. In practice the irreducibility condition does not hold for the web graph. The classical solution for this problem is to add a small positive value to the adjacency matrix [4, 21] in order to get a positive matrix. Tomlin proposed an alternative approach based on the network flow model. We consider the following nonlinear network flow problem with network given by

$$A' = \begin{bmatrix} A & 1 \\ 1^T & 0 \end{bmatrix}$$

where  $1$  denotes the vector with all entries equal to 1.

$$\begin{aligned} \max_{\rho \geq 0} \quad & - \sum_{i,j \in [n+1]} \rho_{i,j} (\log(\frac{\rho_{i,j}}{A'_{i,j}}) - 1) \\ & \sum_{j \in [n+1]} \rho_{i,j} = \sum_{j \in [n+1]} \rho_{j,i} , \quad \forall i \in [n+1] & (p_i) \\ & \sum_{i,j \in [n+1]} \rho_{ij} = 1 & (\mu) \\ & \sum_{j \in [n]} \rho_{n+1,j} = 1 - \alpha & (a) \\ & 1 - \alpha = \sum_{i \in [n]} \rho_{i,n+1} & (b) \end{aligned}$$

We use the conventions that  $0 \log(0) = 0$  and that  $x \log(x/0) = 0$  if and only if  $x = 0$ . In this new model, we add an artificial node connected to all the other nodes and such that the flow through this node is prescribed to be  $1 - \alpha$ .



The algorithm is designed for the minimization of the dual function  $\theta$  where

$$\begin{aligned} \theta(p, \mu, a, b) = & \sum_{i,j \in [n]} A_{ij} e^{p_i - p_j + \mu} + \sum_{i \in [n]} e^{-b - p_{n+1} + p_i + \mu} \\ & + \sum_{j \in [n]} e^{a + p_{n+1} - p_j + \mu} - (1 - \alpha)a - \mu + (1 - \alpha)b . \end{aligned} \quad (4.1)$$

We first give the following counter-example, showing that the problem may be ill posed.

COUNTER-EXAMPLE 1. *The dual function  $\theta$  may be unbounded.*

*Proof.* Take

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

We have

$$\tilde{\theta}(p) = C(\alpha) + (1 - \alpha) \left( \log \left( \sum_{i \in [n]} e^{p_i} \right) + \log \left( \sum_{i \in [n]} e^{-p_i} \right) \right) + (2\alpha - 1) \log(e^{p_1 - p_2} + e^{p_2 - p_3})$$

where  $C(\alpha) \in \mathbb{R}$ . For all  $k \in \mathbb{R}$ ,

$$\begin{aligned} \tilde{\theta}(-k, 0, k, 0) &= C(\alpha) + 2(1 - \alpha) \log(1 + e^k + e^{-k}) + (2\alpha - 1) \log(e^{-k} + e^{-k}) \\ &= C(\alpha) + 2(1 - \alpha)k - (2\alpha - 1)k + 2(1 - \alpha) \log(1 + e^{-k} + e^{-2k}) + (2\alpha - 1) \log(2) \end{aligned}$$

For  $\alpha > \frac{3}{4}$ ,  $\theta$  is unbounded.  $\square$

This example is indeed rather degenerate: the HOTS algorithm can only diverge because it searches the minimum of an unbounded function. Said otherwise, it tries to solve a network flow problem without any admissible flow. We shall give conditions under which there exists a HOTS vector and show that the HOTS algorithm converges to the HOTS vector when these conditions hold.

REMARK 4. *A natural idea to establish the convergence of a fixed point algorithm is to show that it is a contraction in Hilbert metric. In the case of the effective HOTS algorithm, even when the matrix  $A$  is positive, the fixed point algorithm may not be a contraction (take a perturbation of Counter-example 1).*

LEMMA 4.1 ([31]). *For any  $p \in \mathbb{R}^{n+1}$ , the minimum of  $\theta(p, \mu, a, b)$  with respect to  $\mu$ ,  $a$  and  $b$  is unique and given by*

$$\begin{aligned} \mu &= \log \left( \frac{2\alpha - 1}{\sum_{i,j \in [n]} A_{ij} e^{p_i - p_j}} \right) , \\ a &= \log \left( \frac{1 - \alpha}{2\alpha - 1} \frac{\sum_{i,j \in [n]} A_{ij} e^{p_i - p_j}}{\sum_{j \in [n]} e^{p_{n+1} - p_j}} \right) , \\ b &= -\log \left( \frac{1 - \alpha}{2\alpha - 1} \frac{\sum_{i,j \in [n]} A_{ij} e^{p_i - p_j}}{\sum_{i \in [n]} e^{p_i - p_{n+1}}} \right) . \end{aligned}$$

*Proof.* The function  $\theta(p, \cdot, \cdot, \cdot)$  is convex and differentiable so the optimality condition is just that the gradient is zero. One can easily see that the only triple that cancels the gradient is the one given in the lemma.  $\square$

We denote  $\lambda = (\mu, a, b)$  and  $\lambda(p)$  the solution of the minimization of  $\theta(p, \lambda)$  with respect to  $\lambda$ . For  $\lambda \in \mathbb{R}^3$ , we denote

$$f_i^\lambda(p) = \frac{1}{2}(\log(\sum_{j \in [n]} A_{j,i} e^{p_j} + e^{p_{n+1}+a}) - \log(\sum_{k \in [n]} A_{i,k} e^{-p_k} + e^{-p_{n+1}-b})) . \quad (4.2)$$

We also define  $g^\lambda = \exp \circ f^\lambda \circ \log$ :

$$g_i^\lambda(y) = \left( \frac{\sum_{j \in [n]} A_{j,i} y_j + e^a y_{n+1}}{\sum_{k \in [n]} A_{i,k} (y_k)^{-1} + e^{-b} (y_{n+1})^{-1}} \right)^{\frac{1}{2}} .$$

---

**Algorithm 5** Effective HOTS algorithm

---

Given an initial point  $y^0 \in \mathbb{R}^n$ , the effective HOTS algorithm is defined by

$$y^{k+1} = g^{\lambda(\log(y^k))}(y^k)$$


---

LEMMA 4.2 ([9]). *Let  $d \in \mathbb{R}^n$  such that for all  $i$ ,  $d_i > 0$ . The function  $\tilde{\theta}$  defined by  $\tilde{\theta}(p) = \min_{p_{n+1}, \mu, a, b} \theta((p, p_{n+1}), \mu, a, b)$  is strictly convex on the hyperplane  $H = \{x \in \mathbb{R}^n \mid \sum_{i \in [n]} d_i x_i = 0\}$ . In particular, there exists at most one HOTS vector up to an additive constant.*

*Proof.* From the expressions of  $a + p_{n+1}$ ,  $b + p_{n+1}$  and  $\mu$  at the optimum (Lemma 4.1), respectively given by  $e^{a+p_{n+1}} = \frac{(1-\alpha)e^{-\mu}}{\sum_{j \in [n]} e^{-p_j}}$ ,  $e^{-b-p_{n+1}} = \frac{(1-\alpha)e^{-\mu}}{\sum_{i \in [n]} e^{p_i}}$  and  $e^\mu = \frac{2\alpha-1}{\sum_{i,j \in [n]} A_{ij} e^{p_i-p_j}}$ , we can write  $\tilde{\theta}$  as

$$\tilde{\theta}(p) = C(\alpha) + (1-\alpha)\phi(-p) + (1-\alpha)\phi(p) + (2\alpha-1)\log(\sum_{i,j \in [n]} A_{i,j} e^{p_i-p_j})$$

where  $\phi : x \mapsto \log(\sum_{i \in [n]} e^{x_i})$  is the log-sum-exp function, which is strictly convex on any hyperplane that does not containing the vector with all entries equal to 1, and  $C(\alpha) = 1 - 2(1-\alpha)\log(1-\alpha) - (2\alpha-1)\log(2\alpha-1)$ .  $\tilde{\theta}$  is the sum of convex functions and strictly convex functions, so it is strictly convex on  $H$ . We conclude that the minimum of  $\tilde{\theta}$  on  $H$  is unique if it exists. We can then extend this result to the whole space since  $\theta(\eta + p) = \theta(p)$  for all real number  $\eta$ .  $\square$

LEMMA 4.3. *Let  $d$  and  $\tilde{\theta}$  be as in Lemma 4.2. The function  $\tilde{\theta}$  is coercive on the hyperplane  $\{x \in \mathbb{R}^{n+1} \mid \sum_{i \in [n+1]} d_i x_i = 0\}$  if and only if there exists a primal solution with the same pattern as  $A'$ .*

*Proof.* If the function  $\theta$  is coercive, there exists a dual solution and thus there also exists a primal solution with the same pattern as  $A'$ .

If there exists a primal solution with the same pattern as  $A'$ , the constraint qualification conditions are satisfied [27], and there exists a dual solution. By Lemma 4.2,  $\tilde{\theta}$  is strictly convex on the hyperplane  $\{x \in \mathbb{R}^{n+1} \mid \sum_{i \in [n+1]} d_i x_i = 0\}$ . Thus it is necessarily coercive on this hyperplane.  $\square$

LEMMA 4.4. *If  $A \neq 0$ , then for any fixed  $\lambda$ , the iterative algorithm consisting in successive applications of the map  $f^\lambda$ , defined in (4.2), converges to a minimizer of the function  $(p \mapsto \theta(p, \lambda))$ . Moreover, this minimizer is unique up to an additive constant.*

*Proof.* The map  $f^\lambda$  corresponds to the ideal HOTS fixed point operator (3.1) for the matrix

$$\begin{bmatrix} A & e^a \mathbf{1} \\ e^b \mathbf{1}^T & 0 \end{bmatrix}$$

where  $\mathbf{1}$  denotes the vector with all entries equal to 1. As this matrix is primitive as soon as  $A \neq 0$ , Theorem 3.6 and Proposition 3.2 apply.  $\square$

**THEOREM 4.5.** *Let  $F$  defined by  $F(p) = f^{\lambda(p)}(p)$  as in (4.2). Let  $p^*$  be the logarithm of a HOTS vector defined by  $p^* = F(p^*)$ . The matrix  $\nabla F(p^*)$  has all its eigenvalues in the real interval  $(-1, 1]$  and the eigenvalue 1 is simple.*

*Proof.* Let us denote  $\gamma = \frac{1-\alpha}{2\alpha-1}$ ,  $e^a = \gamma \frac{\sum_{i,j \in [n]} A_{ij} e^{p_i - p_j}}{\sum_j e^{-p_j}}$ ,  $e^{-b} = \gamma \frac{\sum_{i,j \in [n]} A_{ij} e^{p_i - p_j}}{\sum_i e^{p_i}}$ . Then for all  $k \in [n]$ ,

$$\begin{aligned} F_k(p) &= \frac{1}{2} \log\left(\sum_{i \in [n]} A_{ik} e^{p_i} + e^a\right) - \frac{1}{2} \log\left(\sum_{j \in [n]} A_{kj} e^{-p_j} + e^{-b}\right) \\ F_{n+1}(p) &= \frac{1}{2} \log\left(\sum_{i \in [n]} e^{p_i}\right) - \frac{1}{2} \log\left(\sum_{j \in [n]} e^{-p_j}\right). \end{aligned}$$

As no coordinate of  $F(p)$  depends on  $p_{n+1}$ , we may consider the reduced function that we shall still denote  $F$  and such that to  $p \in \mathbb{R}^n$  associates  $F(p, 0)$ . The eigenvalues of the gradient of original function are 0 and the eigenvalues of the gradient of the reduced function.

First, we have

$$F(p) = p + \frac{1}{2} \log(1 - \text{diag}(d) \frac{\partial \tilde{\theta}}{\partial p})$$

where  $\tilde{\theta}(p) = \min_{p_{n+1}, \mu, a, b} \theta((p, p_{n+1}), \mu, a, b)$  and  $d_l = \frac{1}{2\alpha-1} \frac{e^{p_l} (\sum_{i,j \in [n]} A_{ij} e^{p_i - p_j})}{\sum_{i \in [n]} A_{il} e^{p_i} + e^a} > 0$  (see [9] for more details on this calculus). Differentiating this equality, we deduce that  $\nabla F = I_n - \frac{1}{2} \text{diag}(d) \nabla^2 \tilde{\theta}$ . Let  $\lambda$  be an eigenvalue of  $\nabla F$ . This means that there exist a vector  $x$  such that

$$\lambda x = \nabla F x = x - \frac{1}{2} \text{diag}(d) \nabla^2 \tilde{\theta} x$$

$$\nabla^2 \tilde{\theta} x = 2(1 - \lambda) \text{diag}(d^{-1}) x$$

This is a generalized eigenvalue problem with  $\nabla^2 \tilde{\theta}$  symmetric semi-definite positive by convexity of  $\tilde{\theta}$  and  $\text{diag}(d^{-1})$  diagonal definite positive. Hence  $2(1 - \lambda)$  is necessarily a nonnegative real number and  $\lambda$  is real and smaller than 1. Also, if  $\lambda = 1$ , this means that  $x$  is the vector with all its entries equal to 1 (by Proposition 10 in [9] which is a simple extension of Lemma 4.2) and thus  $\lambda$  is simple.

We shall now show that all the eigenvalues of  $\nabla F(p^*)$  are stricly greater than  $-1$ . Differentiating the expression of  $a$ , we get

$$\frac{\partial e^a}{\partial p_l}(p) = \gamma \left( -\frac{\sum_{i \in [n]} A_{il} e^{p_i - p_l}}{\sum_{j \in [n]} e^{-p_j}} + \frac{\sum_{j \in [n]} A_{lj} e^{p_l - p_j}}{\sum_{j \in [n]} e^{-p_j}} + \frac{\sum_{i,j \in [n]} A_{ij} e^{p_i - p_j}}{(\sum_{j \in [n]} e^{-p_j})^2} e^{-p_l} \right)$$

But as  $p^*$  is a fixed point of  $F$ , it satisfies the equality

$$\sum_{i \in [n]} A_{il} e^{p_i^* - p_l^*} + e^{a^* - p_l^*} = \sum_{j \in [n]} A_{lj} e^{p_l^* - p_j^*} + e^{-b^* + p_l^*}$$

which can be rewritten as

$$\sum_{j \in [n]} A_{lj} e^{p_l^* - p_j^*} - \sum_{i \in [n]} A_{il} e^{p_i^* - p_l^*} = \gamma \sum_{i, j \in [n]} A_{ij} e^{p_i^* - p_j^*} \left( \frac{e^{-p_l^*}}{\sum_{j \in [n]} e^{-p_j^*}} - \frac{e^{p_l^*}}{\sum_{i \in [n]} e^{p_i^*}} \right).$$

Hence

$$\frac{\partial e^a}{\partial p_l}(p^*) = \gamma \sum_{i, j \in [n]} A_{ij} e^{p_i^* - p_j^*} \left( \gamma \frac{e^{-p_l^*}}{(\sum_j e^{-p_j^*})^2} - \gamma \frac{e^{p_l^*}}{\sum_i e^{p_i^*} \sum_j e^{-p_j^*}} + \frac{e^{-p_l^*}}{(\sum_j e^{-p_j^*})^2} \right)$$

Let us introduce  $d'$  such that

$$d'_k{}^{-1} = \sum_{i \in [n]} A_{ik} e^{p_i^* - p_k^*} + e^{a^* - p_k^*} = \sum_{j \in [n]} A_{kj} e^{p_k^* - p_j^*} + e^{-b^* + p_k^*}. \quad (4.3)$$

Doing the same for  $e^{-b}$  as for  $e^a$  and differentiating  $F$ , we get

$$\begin{aligned} \frac{\partial F_k}{\partial p_l}(p^*) &= \frac{1}{2} d'_k A_{l,k} e^{p_l^* - p_k^*} + \frac{1}{2} d'_k A_{k,l} e^{p_k^* - p_l^*} + \frac{1}{2} \gamma \sum_{i, j \in [n]} A_{ij} e^{p_i^* - p_j^*} d'_k \times \\ &\left( (1 + \gamma) \frac{e^{-p_l^* - p_k^*}}{(\sum_j e^{-p_j^*})^2} - \gamma \frac{e^{p_l^* - p_k^*}}{\sum_i e^{p_i^*} \sum_j e^{-p_j^*}} - \gamma \frac{e^{p_k^* - p_l^*}}{\sum_i e^{p_i^*} \sum_j e^{-p_j^*}} + (1 + \gamma) \frac{e^{p_l^* + p_k^*}}{(\sum_i e^{p_i^*})^2} \right). \end{aligned}$$

We can now decompose  $\frac{\partial F}{\partial p}$  as

$$\frac{\partial F}{\partial p} = D' S + D' R$$

where  $D' = \text{diag}(d')$ ,  $S$  is a symmetric matrix with nonnegative entries

$$S_{k,l} = \frac{1}{2} A_{k,l} e^{p_l - p_k} + \frac{1}{2} A_{l,k} e^{p_k - p_l} + \frac{1}{2} \gamma \sum_{i, j} A_{ij} e^{p_i - p_j} \left( \frac{e^{-p_l - p_k}}{(\sum_j e^{-p_j})^2} + \frac{e^{p_l + p_k}}{(\sum_i e^{p_i})^2} \right)$$

and  $R$  is the following symmetric rank 1 matrix

$$R_{k,l} = \frac{1}{2} \gamma^2 \sum_{i, j} A_{ij} e^{p_i - p_j} \left( \frac{e^{-p_k}}{\sum_j e^{-p_j}} - \frac{e^{p_k}}{\sum_i e^{p_i}} \right) \left( \frac{e^{-p_l}}{\sum_j e^{-p_j}} - \frac{e^{p_l}}{\sum_i e^{p_i}} \right).$$

The nonnegative matrix  $D' S$  verifies that for all  $k$ ,  $\sum_l d'_k S_{k,l} = 1$ , thus by the Perron-Frobenius theorem [2], we have exhibited a Perron vector and the spectral radius of the matrix is 1. Moreover,  $D' S$  is positive, so every other of its eigenvalues has a modulus strictly smaller than 1.

The matrix  $D' R$  is a rank 1 matrix and its only nonzero eigenvalue is positive. Indeed it is equal to  $\frac{1}{2} \gamma^2 \sum_{i, j} A_{ij} e^{p_i - p_j} \sum_k \left( \frac{e^{-p_k}}{\sum_j e^{-p_j}} - \frac{e^{p_k}}{\sum_i e^{p_i}} \right)^2 d'_k$ .

Let  $\lambda$  be an eigenvalue of  $\frac{\partial F}{\partial p} = D'S + D'R$ . It then also an eigenvalue of the similar matrix  $(D')^{1/2}S(D')^{1/2} + (D')^{1/2}R(D')^{1/2}$ , which is symmetric. Hence,

$$\lambda \geq \min_{x \in \mathbb{R}^n: \|x\|_2=1} x^T (D')^{1/2} S (D')^{1/2} x + x^T (D')^{1/2} R (D')^{1/2} x$$

As the spectral radius of  $D'S$  is 1, the same is true for  $(D')^{1/2}S(D')^{1/2}$  and for all vector  $x$ ,  $x^T (D')^{1/2} S (D')^{1/2} x > -\|x\|_2^2$ . As  $D'R$  has only nonnegative eigenvalues,  $(D')^{1/2}R(D')^{1/2}$  is semi-definite positive and  $x^T (D')^{1/2} R (D')^{1/2} x \geq 0$  for all  $x$ . As a conclusion,

$$\lambda \geq \min_{x \in \mathbb{R}^n: \|x\|_2=1} x^T (D')^{1/2} S (D')^{1/2} x + x^T (D')^{1/2} R (D')^{1/2} x > -1 . \quad \square$$

**THEOREM 4.6.** *Let  $F(p) = f^{\lambda(p)}(p)$  as in (4.2). If there exists a primal feasible point with the same pattern as  $A'$ , then the effective HOTS algorithm (Algorithm 5) converges to a HOTS vector  $e^{p^*}$  (unique up to a multiplicative constant) linearly at a rate  $|\lambda_2(\nabla F(p^*))| = \max\{|\lambda|; \lambda \in \text{spectrum}(\nabla F(p^*)), \lambda \neq 1\}$ ,*

*Proof.* Let  $\tilde{F}$  be the map defined by  $\tilde{F}(p) = F(p) - \frac{1}{1^T d'^{-1}} \sum_{i \in [n+1]} (d'_i)^{-1} F_i(p)$ , with  $d'_i$  defined in (4.3) in the proof of Theorem 4.5 for  $i \in [n]$  and  $(d'_{n+1})^{-1} = 0$ . For all  $k$ , let  $p_k$  be the  $k$ th iterate of the HOTS algorithm, i.e.  $p_{k+1} = F(p_k)$ , and let  $\lambda_k = \lambda(p_k)$ . We also define  $q_k$  by  $q_0 = p_0$  and  $q_{k+1} = \tilde{F}(q_k)$ . By Theorem 3.9 and by definition of  $\lambda_{k+1}$ , we have  $\theta(p_k, \lambda_k) \geq \theta(p_{k+1}, \lambda_k) \geq \theta(p_{k+1}, \lambda_{k+1})$ . As  $q_k - p_k$  is proportional to the vector with all entries equal to 1,  $\lambda(p_k) = \lambda(q_k) = \lambda_k$  and  $\theta(q_k, \lambda) = \theta(p_k, \lambda)$  for all  $k$  and  $\lambda$ . Hence

$$\theta(q_k, \lambda_k) \geq \theta(q_{k+1}, \lambda_k) \geq \theta(q_{k+1}, \lambda_{k+1}) . \quad (4.4)$$

Now, for all  $k$ ,  $q_k \in H = \{x \in \mathbb{R}^{n+1} \mid \sum_{i \in [n+1]} (d'_i)^{-1} x_i = 0\}$ . As by Lemma 4.3,  $\tilde{\theta}$  is coercive on  $H$ ,  $\theta$  is bounded from below and  $(\theta(q_k, \lambda_k))_k$  converges to, say,  $\bar{\theta}$ . Moreover, the sequence  $(q_k)_k$  must be bounded. By continuity of the function  $\lambda(\cdot)$ ,  $(\lambda_k)_k$  is also bounded. Hence, they have limit points.

Let  $\bar{q}$  be a limit point of  $(q_k)_k$  and  $\bar{\lambda} = \lambda(\bar{q})$ . For all  $\epsilon > 0$  and  $K > 0$ , there exists  $k \geq K$  and  $k' \geq k + 1$  such that  $\|q_k - \bar{q}\| \leq \epsilon$ ,  $\|q_{k'} - \bar{q}\| \leq \epsilon$ . By (4.4),

$$\theta(q_k, \lambda_k) \geq \theta(q_{k+1}, \lambda_k) \geq \theta(q_{k+1}, \lambda(q_{k+1})) \geq \theta(q_{k'}, \lambda_{k'})$$

where  $q_{k+1} = \tilde{F}(q_k)$ . When  $\epsilon$  tends to 0 and  $K$  tends to infinity, we get with  $\bar{q} = \tilde{F}(\bar{q})$ ,

$$\theta(\bar{q}, \bar{\lambda}) \geq \theta(\bar{q}, \bar{\lambda}) \geq \theta(\bar{q}, \lambda(\bar{q})) \geq \theta(\bar{q}, \bar{\lambda}) .$$

In particular,  $\theta(\bar{q}, \bar{\lambda}) = \theta(\bar{q}, \lambda(\bar{q}))$ . This implies by Lemma 4.1 that  $\bar{\lambda} \in \arg \min_{\lambda} \theta(\bar{q}, \lambda)$  and thus  $\bar{\lambda} = \lambda(\bar{q})$  by uniqueness of the minimizer.

Similarly,  $\bar{q}$  is also a limit point of  $(q_k)_k$  and we may consider the sequence  $(u_k)_k$  such that  $u_k = (\tilde{F})^k(\bar{q}) = (f^{\bar{\lambda}})^k(\bar{q}) - \frac{1}{1^T d'^{-1}} \sum_{i \in [n+1]} d'_i{}^{-1} ((f^{\bar{\lambda}})^k(\bar{q}))_i$ . Iterating the argument of the preceeding paragraph, for any  $k$ ,  $\lambda(u_k) = \bar{\lambda}$  and  $u_k$  is a limit point of  $(q_k)_k$ . Now, by Lemma 4.4, the sequence  $(u_k)$  converges to  $q^* \in \arg \min_q \theta(q, \bar{\lambda})$ . As we also have  $\lambda(q^*) = \bar{\lambda}$ , we conclude that  $(q^*, \bar{\lambda})$  is a minimizer of  $\theta$  and that there exists a limit point of  $(q_k, \lambda_k)_k$  that minimizes  $\theta$ . Now, as  $(\theta(q_k, \lambda_k))_k$  is decreasing, all the limit points of  $(q_k)$  minimize  $\theta$ . The uniqueness of the minimizer of  $\theta$  on  $H$  (Lemma 4.2) gives the convergence of the effective HOTS algorithm to the HOTS vector in the projective space, that is the convergence of the sequence  $(q_k)_k$ .

We shall now prove that the sequence  $(p_k)_{k \geq 0}$  indeed converges. Let us denote by  $\rho = \max\{|\lambda|; \lambda \in \text{spectrum}(\nabla F(q^*)), \lambda \neq 1\}$ . By Theorem 4.5, we know  $\rho < 1$ . Denoting  $(\lambda_i, u_i, v_i)_{i \in [n+1]}$ , the eigenvalues and eigenvectors of  $\nabla F(q^*) = \sum_{i=1}^{n+1} \lambda_i u_i v_i^T$ , we have  $\nabla \tilde{F}(q^*) = \sum_{i=2}^{n+1} \lambda_i u_i v_i^T + \frac{1(d'^{-1})^T}{1^T d'^{-1}} - \frac{1}{1^T d'^{-1}} (d'^{-1})^T \frac{1(d'^{-1})^T}{1^T d'^{-1}} = \sum_{i=2}^{n+1} \lambda_i u_i v_i^T$ . Hence by [25], for all  $\epsilon' > 0$ , there exists a norm  $\|\cdot\|$  such that for all  $x \in \mathbb{R}^{n+1}$ ,  $\|\nabla \tilde{F}(q^*)x\| = \|\sum_{i=2}^n \lambda_i u_i v_i^T x\| \leq (\rho + \epsilon')\|x\|$ .

Thus for  $x \in \mathbb{R}^{n+1}$  sufficiently close to  $q^*$ ,

$$\|\tilde{F}(x) - q^*\| = \|\tilde{F}(x) - \tilde{F}(q^*)\| \leq (1 + \epsilon'/2)\|\nabla F(q^*)(x - q^*)\| \leq (\rho + \epsilon')\|x - q^*\|.$$

We deduce that  $(q_k)$  converges linearly at rate  $\rho$  to  $q^*$ . Now for all  $k$ , we have

$$\begin{aligned} p_k &= q_k + \sum_{l=0}^{k-1} \frac{1}{1^T d'^{-1}} \sum_{i \in [n+1]} d'_i{}^{-1} F_i(q_l) \\ &= q_k + \sum_{l=0}^{k-1} \frac{1}{1^T d'^{-1}} \sum_{i \in [n+1]} d'_i{}^{-1} F_i(q_l) - F_i(q^*) = q_k + \sum_{l=0}^{k-1} \eta_l, \end{aligned}$$

where  $|\eta_l| = O(\|F(q_l) - F(q^*)\|) = O(\|q_l - q^*\|) = O(\rho^l)$ . Hence  $\sum_{l=0}^{k-1} \eta_l$  is summable and converges linearly at rate  $\rho$ . Finally,  $(p_k)$  converges linearly at rate  $\rho$ . Like in the proof of Theorem 4.6 we deduce the convergence of the sequence  $(\exp(p_k))$  linearly at rate  $\rho$  to a HOTS vector.  $\square$

The last result shows that coordinate descent is an alternative algorithm for the computation of the effective HOTS vector.

**PROPOSITION 4.7.** *If there exists a primal feasible point with the same pattern as  $A'$ , the coordinate descent algorithm applied to the unrestricted minimization of the dual function  $\theta$  defined in (4.1) and choosing coordinates in a cyclic order converges linearly to a HOTS vector.*

*Proof.* If there exists a primal feasible point with the same pattern as  $A'$ , then the set of minimizers of  $\theta$  is nonempty by Lemmas 4.2 and 4.3. The function  $\theta$  has the required form with  $\psi(x) = \sum_{i,j} A_{i,j} \exp(x_{i,j})$ . The hessian of  $\psi$  is clearly definite positive for all  $x$ . Thus the hypotheses of Proposition 3.1 are verified and the result follows.  $\square$

## 5. An exact coordinate descent for the truncated scaling problem.

Truncated scaling problems were introduced by Schneider in [27, 28] in order to generalize both matrix balancing and row-column equivalence scaling. Given a  $n \times n$  matrix  $A$  and bounds  $L$  and  $U$  such that  $L_{i,j} \leq U_{i,j}$ , the truncated scaling problem consists in finding a matrix  $X$  of the form  $X = T(D^{-1}AD)$  with  $D$  diagonal definite positive,  $T$  the truncation operator  $T_{i,j}(X) = \max(\min(U_{i,j}, X_{i,j}), L_{i,j})$  and such that  $\sum_k X_{i,k} = \sum_j X_{j,i}$  for all  $i$ . This problem is equivalent to the following optimization problem

$$\begin{aligned} \max_{\rho \geq 0} \quad & - \sum_{i,j \in [n]} \rho_{i,j} (\log(\frac{\rho_{i,j}}{A_{i,j}}) - 1) \\ \sum_{j \in [n]} \rho_{i,j} &= \sum_{j \in [n]} \rho_{j,i}, \quad \forall i \in [n] & (p_i) \\ L_{i,j} \leq \rho_{i,j} &\leq U_{i,j}, \quad \forall i, j \in [n] & (\eta_{i,j}, \zeta_{i,j}) \end{aligned}$$

If the bounds satisfy  $L_{i,j} = 0$  and  $U_{i,j} = +\infty$  for all  $i, j$ , then we have a matrix balancing problem. The reduction of row-column equivalence scaling to truncated scaling lies in a graph transformation described by Schneider in [27], Lemma 1.

The dual function can take two forms. In [27], Schneider proposes to relax the equality constraints and to let the bound constraints in the objective function. One gets a dual function of the form

$$\Psi(p) = \sum_{i,j \in [n]} \psi_{i,j}^*(p_i - p_j) \quad (5.1)$$

where each  $\psi_{i,j}^* : \mathbb{R} \rightarrow \mathbb{R}$  is convex. Then one can perform an inexact coordinate descent where at each step the minimization along the coordinate is not necessarily exact.

Here, we shall study the choice of relaxing all the constraints. This approach has been proposed in [32] for another generalization of the row-column equivalence scaling problem (but this generalization does not include truncated scaling). Then, we get the following dual function:

$$\theta(p, \eta, \zeta) = \sum_{i,j \in [n]} \phi_{i,j}^*(p_i - p_j + \eta_{i,j} - \zeta_{i,j}) - \sum_{i,j \in [n]} L_{i,j} \eta_{i,j} + \sum_{i,j \in [n]} U_{i,j} \zeta_{i,j} \quad , \quad (5.2)$$

where  $\phi_{i,j}^*(t) = A_{i,j} e^t$ . We shall minimize  $\theta$  with unrestricted  $p$  and nonnegative  $\eta$  and  $\zeta$ . As in [6, 32], we shall show in Proposition 5.1 that exact expressions of the minimizers along one single coordinate exist.

**PROPOSITION 5.1.** *Given  $p \in \mathbb{R}^n$ , the minimizers of  $\min_{\eta \geq 0, \zeta \geq 0} \theta(p, \eta, \zeta)$  are given for all  $i$  and  $j$  by*

$$\begin{aligned} \exp(\eta_{i,j}) &= \max\left(\frac{L_{i,j}}{A_{i,j} e^{p_i - p_j}}, 1\right) \\ \exp(-\zeta_{i,j}) &= \min\left(\frac{U_{i,j}}{A_{i,j} e^{p_i - p_j}}, 1\right) \end{aligned}$$

*Proof.* The proofs for  $\eta$  and  $\zeta$  are symmetric, so we only do the one for  $\eta$ .

$$\frac{\partial \theta}{\partial \eta_{i,j}} = A_{i,j} e^{p_i - p_j + \eta_{i,j} - \zeta_{i,j}} - L_{i,j}$$

Two cases may occur: either  $\frac{\partial \theta}{\partial \eta_{i,j}} = 0$  and  $\eta_{i,j} \geq 0$  or  $\eta_{i,j} = 0$  and  $\frac{\partial \theta}{\partial \eta_{i,j}} \geq 0$ . Hence, either  $\exp(\eta_{i,j}) = \frac{L_{i,j}}{A_{i,j} e^{p_i - p_j - \zeta_{i,j}}}$  and  $\exp(\eta_{i,j}) \geq 1$  or we have  $\exp(\eta_{i,j}) = 1$  and  $1 \geq \frac{L_{i,j}}{A_{i,j} e^{p_i - p_j - \zeta_{i,j}}} \geq \frac{L_{i,j}}{A_{i,j} e^{p_i - p_j}}$ . We thus have the result if  $\eta_{i,j}$  and  $\zeta_{i,j}$  are not positive together.

Now suppose that  $\eta_{i,j} > 0$  and  $\zeta_{i,j} > 0$ . In this case, we have  $\exp(\eta_{i,j}) = \frac{L_{i,j}}{A_{i,j} e^{p_i - p_j - \zeta_{i,j}}}$  and  $\exp(-\zeta_{i,j}) = \frac{U_{i,j}}{A_{i,j} e^{p_i - p_j + \eta_{i,j}}}$ . This implies that  $U_{i,j} = L_{i,j}$ . Thus the two bound constraints are in fact an equality constraint and we shall consider the unconstrained multiplier  $\xi_{i,j} = \eta_{i,j} - \zeta_{i,j}$  instead of the two former multipliers. Then  $\xi_{i,j}$  verifies  $\exp(\xi_{i,j}) = \frac{L_{i,j}}{A_{i,j} e^{p_i - p_j}}$ . It is unique and can be redecomposed into  $\zeta_{i,j}$  and  $\eta_{i,j}$  as in the proposition.  $\square$

This proposition shows that we do not need to store the values of  $\eta_{i,j}$  and  $\zeta_{i,j}$ . Indeed, for fixed  $p$  and  $\lambda$ , if  $(\eta, \zeta) = \arg \min_{\eta' \geq 0, \zeta' \geq 0} \theta(p, \lambda, \eta', \zeta')$ , then if we denote

by  $\text{mid}(x, a, b) = \max(\min(x, a), b)$

$$\begin{aligned} A_{i,k} e^{p_i - \eta_{i,k} + \zeta_{i,k}} &= \text{mid}(A_{i,k} e^{p_i}, U_{i,k} e^{p_k}, L_{i,k} e^{p_k}) \\ A_{k,j} e^{-p_j - \eta_{k,j} + \zeta_{k,j}} &= \text{mid}(A_{k,j} e^{-p_j}, U_{k,j} e^{-p_k}, L_{k,j} e^{-p_k}) . \end{aligned}$$

This gives an expression of coordinate descent with no storage of  $\eta_{k+1}$  nor  $\zeta_{k+1}$ .

---

**Algorithm 6** Exact coordinate descent for Truncated scaling

---

Given an initial vector  $p^0$ , calculate iteratively  $p^k$  by selecting a coordinate  $l$  following a cyclic rule and computing:  $p_{l'}^{k+1} = p_{l'}^k$  for all  $l' \neq l$  and

$$\begin{aligned} p_l^{k+1} &= \frac{1}{2} \log \left( \sum_{i \in [n]} \text{mid}(A_{i,l} e^{p_i^k}, U_{i,l} e^{p_l^k}, L_{i,l} e^{p_l^k}) \right) \\ &\quad - \frac{1}{2} \log \left( \sum_{j \in [n]} \text{mid}(A_{l,j} e^{-p_j^k}, U_{l,j} e^{-p_l^k}, L_{l,j} e^{-p_l^k}) \right) \end{aligned}$$


---

**PROPOSITION 5.2.** *If  $A$  has a truncated scaling, then Algorithm 6 converges linearly to a solution of the truncated scaling problem.*

*Proof.* By Proposition 5.1, we can see that Algorithm 6 is a coordinate descent algorithm (Algorithm 3) applied to the minimization of the dual function  $\theta$  defined in (5.2) such that for every  $l$ , the coordinate selection order is

$$\eta_{l,1}, \dots, \eta_{l,n}, \zeta_{l,1}, \dots, \zeta_{l,n}, p_l$$

As in Proposition 4.7, we then just verify the hypotheses of Proposition 3.1.  $\square$

**REMARK 5.** *Due to the truncation, it is not clear how to determine the primitivity of the gradient of the fixed point operator of a HOTS-like algorithm for the truncated scaling problem.*

Tomlin proposed in [31] to search for a flow of websurfers  $\rho$  that maximizes the entropy for the effective HOTS problem (Section 4) with additional bound constraints of the type

$$U_{i,j} \leq \rho_{i,j} \leq L_{i,j} . \quad (5.3)$$

These bound constraints model the fact that one may have information on the actual flow of websurfers through some hyperlink, even if the flow on every hyperlink is out of reach.

We propose the following coordinate descent algorithm (Algorithm 7) that couples the algorithms presented in Proposition 4.7 and Algorithm 6. The proof of convergence is just the concatenation of the arguments of Propositions 4.7 and 5.2.

**6. Comparison with alternative algorithms.** We give in Table 6.1 a comparison of four algorithms for the matrix balancing problem: interior-reflective Newton method (Matlab `fminunc` function), coordinate descent, DomEig and ideal HOTS. We considered a small matrix, a medium size matrix and two large matrices. The matrix  $A = \begin{bmatrix} \epsilon & 1 \\ 2 & 0 \end{bmatrix}$ , with  $\epsilon = 10^{-3}$ , is a nearly imprimitive matrix. The CMAP matrix is the



---

**Algorithm 7** Coordinate descent for the HOTS problem with bounds

---

Start with an initial point  $y^0 \in \mathbb{R}^n$ ,  $y^0 > 0$ . Given  $y^k$ , select a coordinate  $i \in [n+1]$  and compute  $y^{k+1}$  such that  $y_j^{k+1} = y_j^k$ ,  $\forall j \neq i$  and

$$y_i^{k+1} = \left( \frac{\sum_{i \in [n]} \text{mid}(A_{i,l} y_i^k, U_{i,l} y_l^k, L_{i,l} y_l^k) + e^{a^k} y_{n+1}^k}{\sum_{j \in [n]} \text{mid}(A_{l,j} (y_j^k)^{-1}, U_{l,j} (y_l^k)^{-1}, L_{l,j} (y_l^k)^{-1}) + e^{-b^k} (y_{n+1}^k)^{-1}} \right)^{\frac{1}{2}}$$

If  $i < n+1$ , then set  $(\mu^{k+1}, a^{k+1}, b^{k+1}) = (\mu^k, a^k, b^k)$ , otherwise

$$\begin{aligned} \mu^{k+1} &= \log\left(\frac{2\alpha - 1}{\sum_{j,j' \in [n]} \text{mid}(A_{j,j'} \frac{y_j^k}{y_{j'}^k}, U_{j,j'}, L_{j,j'})}\right), \\ a^{k+1} &= \log\left(\frac{1 - \alpha}{\sum_{j \in [n]} \frac{y_{n+1}^k}{y_j^k} e^{\mu^{k+1}}}\right), \quad b^{k+1} = -\log\left(\frac{1 - \alpha}{\sum_{j' \in [n]} \frac{y_{j'}^k}{y_{n+1}^k} e^{\mu^{k+1}}}\right). \end{aligned}$$

---

	$A = \begin{bmatrix} \epsilon & 1 \\ 2 & 0 \end{bmatrix}$	CMAP 1,500 p.	NZ Uni 413,639 p.	uk2002 18,520,486 p.
$ \lambda_2(P) $ (Thm. 3.6)	0.9993	0.8739	0.9774	0.998
Matlab's fminunc	0.015 s	948 s	out of mem.	-
DomEig [14]	1.87 s	34.4 s	> 600 s	-
Coordinate desc. (Alg. 2)	0.006 s	0.03 s	6.06 s	2,391 s
Ideal HOTS (Alg. 1)	2.0 s	0.02 s	7.52 s	1,868 s

---

TABLE 6.1

Execution times for 4 algorithms to solve the matrix balancing problem. General purpose algorithms like Newton methods (fminunc) are outperformed by coordinate descent and ideal HOTS for this problem. DomEig [14] does not seem to be very efficient for these sparse problems. On the other hand, coordinate descent and ideal HOTS have similar performances. We remark however that coordinate descent has a better behavior for imprimitive matrices but that we have the expression of the rate of convergence of ideal HOTS, that we do not have for coordinate descent.

adjacency matrix of a fragment of the web graph of size 1,500. The crawl consists of the [www.cmap.polytechnique.fr](http://www.cmap.polytechnique.fr) website and surrounding pages. The NZ Uni matrix comes from a crawl of New Zealand Universities websites, available at [26]. It has 413,639 pages. The uk2002 matrix comes from a crawl of the .uk name domain with 18,520,486 pages, gathered by UbiCrawler [3]. For the matrix balancing problem, we added to the entries of the adjacency matrices arising from fragments of the web graph a small positive constant equal to  $1/n$  (to guarantee irreducibility of the matrix). We launched our numerical experiments on a personal computer with 4 Intel Xeon CPUs at 2.98 GHz and 8 GB RAM.

Convex optimization solvers, coding algorithms like quasi-Newton, are heavy machineries that can reach quadratic convergence and can handle general problems. They however need complex algorithms: one should not program them by scratch but use robust public codes. Parallel computation is not so trivial and tuning the parameters may be difficult. Second order methods also need to compute the Hessian matrix, which may be a large dense matrix.

The DomEig algorithm [14] consists of 2 loops. The inner loop is the power

	$A = \begin{bmatrix} \epsilon & 1 \\ 2 & 0 \end{bmatrix}$	CMA	NZ Uni	uk2002
$ \lambda_2(\nabla F) $ (Prop. 3.8)	0.8846	0.946	0.995	0.9994
Coordinate descent (Prop. 4.7)	0.0061 s	0.0613 s	35.7 s	3,809 s
Effective HOTS (Alg. 5)	0.0217 s	0.0589 s	36.5 s	3,017 s
PageRank [4]	0.0216 s	0.0195 s	2.9 s	270 s

TABLE 6.2

Comparison of coordinate descent and HOTS for the effective HOTS problem with  $\alpha = 0.9$ . Both algorithms seem to perform equally. In particular, unlike PageRank, the convergence rate deteriorates with the size of the fragment of the web graph considered.

method, it has a linear speed of convergence equal to the spectral gap of the matrix considered. Most of the computational cost lies in the power iterations, since the outer loop is simple. This algorithm is specially designed for matrix balancing and accepts no extension. It does not seem to be very efficient for sparse problem, when compared to HOTS and coordinate descent. Moreover, the precision required for the determination of the principal eigenvalue has a strong impact on the performance of the algorithm.

Schneider’s coordinate descent DSS algorithm for the matrix balancing problem is a very efficient and scalable algorithm. It remains efficient for non primitive matrices.

The ideal HOTS algorithm has a linear speed of convergence equal to the spectral gap of  $P$ . On our experiments, it performs well for medium and large size problems but lacks efficiency for imprimitive matrices. For the fragments of the web graph, it converged in a bit more iterations than coordinate descent but each iteration is a bit simpler. So the execution times are about the same. Its advantage compared to coordinate descent is that its implementation only needs left and right matrix-vector products and element-wise operations (division and square root). Hence, it does not require to compute and store the transpose of the adjacency matrix, which can be useful for web scale applications.

In Table 6.2, we compare coordinate descent and the effective HOTS algorithm for the effective HOTS problem. Both algorithms scale well and have comparable computational costs. However, we remark that the rate of convergence seems to deteriorate when the size of the problem increases. It might be necessary to choose smaller values for  $\alpha$  for large graphs in order to compensate this phenomenon. In the next section, we propose an alternative solution consisting in normalizing the adjacency matrix prior to computing the HOTS score, and hence minimizing a relative entropy instead of the entropy function.

In Table 6.3, we give the execution times for the HOTS problem with bounds (5.3).

**7. A normalized HOTS algorithm for a better convergence rate.** The previous study has shown two drawbacks of Tomlin’s HOTS algorithm. First of all, its convergence rate seems to deteriorate when the size of the problem increases. The second problem concerns manipulations of the HOTS score: when one single page is considered, a very good strategy is to point to no page, and thus make this page a dangling node in the web graph. This comes from the relation of HOTS with the anti-Perron ranking (Remark 3). Indeed, the anti-Perron ranking penalizes bad quality links but also adding any outlink, even a good quality one, diminishes the score of the page where it has been added.

	$A = \begin{bmatrix} \epsilon & 1 \\ 2 & 0 \end{bmatrix}$	CMA	NZ Uni
Inexact coor. descent (Eq. (5.1) and [28])	0.019 s	4.28 s	> 2000 s
Exact coor. descent (Alg. 7)	0.005 s	0.23 s	100 s

TABLE 6.3

Execution times for two algorithms to solve the effective HOTS problem where some arcs have prescribed bounds on their flow [31] (these bounds are determined at random). Exact coordinate descent (Algorithm 7) is faster than inexact coordinate descent. Indeed we do not need any line search for Algorithm 7. By comparison with Table 6.2, we can see that the computational expense is multiplied by less than 4 with the bound constraints.

We now propose a modification of the HOTS algorithm that tackles those two issues. In order to stop penalizing the presence of hyperlinks on a page, we normalize the adjacency matrix, and thus state the entropy optimization problem as a relative entropy optimization problem. Then the Perron ranking reduces to PageRank and the normalized anti-Perron ranking does not penalize the number of links any more.

We also need to address the problem of dangling nodes, in which the normalization of the corresponding row of the adjacency matrix is not defined, and the reducibility of the adjacency matrix. A possibility is to keep on inspiring from the PageRank and consider the Google matrix instead of the normalized adjacency matrix. With this choice, dangling pages are considered to point to every page, which implies that they have a low rank in the normalized HOTS score, when compared to the rank given by PageRank. Instead, we suggest to set Tomlin's effective network model to solve the reducibility problem and add another fictitious node that point to every page and is pointed to by every dangling page.

We end up with the following network flow problem where the  $(n+2) \times (n+2)$  matrix  $M$  is defined by:

$$M_{i,j} = \begin{cases} \frac{A_{i,j}}{\sum_k A_{i,k}} & \text{if } i, j \leq n, \sum_k A_{i,k} \geq 1 \\ 0 & \text{if } i, j \leq n, \sum_k A_{i,k} = 0 \end{cases}$$

$$M = \begin{bmatrix} M_{[n],[n]} & f & 1 \\ 1^T & 0 & 1 \\ 1^T & 1 & 0 \end{bmatrix}$$

and  $f$  is the 0-1 vector such that  $f_i = 1$  if and only if  $i$  is a dangling node.

$$\max_{\rho \geq 0} - \sum_{i,j \in [n+2]} \rho_{i,j} \left( \log\left(\frac{\rho_{i,j}}{M_{i,j}}\right) - 1 \right) \quad (7.1)$$

$$\sum_{i,j \in [n+2]} \rho_{ij} = 1, \quad \sum_{j \in [n+2]} \rho_{i,j} = \sum_{j \in [n+2]} \rho_{j,i}, \quad \forall i \in [n+2]$$

$$\sum_{j \in [n]} \rho_{n+2,j} = 1 - \alpha = \sum_{i \in [n]} \rho_{i,n+2}$$

We call this optimization problem the normalized HOTS problem. The normalized HOTS algorithm is defined in the same way as the effective HOTS algorithm but with  $M$  instead of  $A'$ . As in Lemma 4.1, we define  $\mu' = \log\left(\frac{2\alpha-1}{\sum_{i,j \in [n]} M_{i,j} e^{p_i - p_j}}\right)$ ,

	CMAF	NZ Uni	uk2002
$ \lambda_2(\nabla F) $ (classical HOTS)	0.946	0.995	0.9994
$ \lambda_2(\nabla F) $ (normalized HOTS)	0.906	0.988	0.96
Execution times for Normalized HOTS	0.055 s	46.24 s	752 s

TABLE 7.1

Performances of the normalized HOTS algorithm presented in Section 7. The correlation between the deterioration of the convergence rate of the algorithm, given by the second eigenvalue of the matrix  $\nabla F$ , and the size of the data set does not hold any more. Moreover, on all the tests we performed, the convergence rate remained under 0.99.

$a' = \log(\frac{1-\alpha}{2\alpha-1} \frac{\sum_{i,j \in [n]} M_{i,j} e^{p_i - p_j}}{\sum_{j \in [n]} e^{p_{n+1} - p_j}})$ ,  $b' = -\log(\frac{1-\alpha}{2\alpha-1} \frac{\sum_{i,j \in [n]} M_{i,j} e^{p_i - p_j}}{\sum_{i \in [n]} e^{p_i - p_{n+1}}})$  and the triple  $\lambda'(p) = (\mu', a', b')$ . For  $\lambda' \in \mathbb{R}^3$ , we denote

$$g_i^{\lambda'}(y) = \left( \frac{\sum_{j \in [n]} M_{j,i} y_j + e^{a'} y_{n+1}}{\sum_{k \in [n]} M_{i,k} (y_k)^{-1} + e^{-b'} (y_{n+1})^{-1}} \right)^{\frac{1}{2}}.$$

---

**Algorithm 8** Normalized HOTS algorithm

---

Given an initial point  $y^0 \in \mathbb{R}^n$ , the effective HOTS algorithm is defined by

$$y^{k+1} = g^{\lambda'(\log(y^k))}(y^k)$$


---

**PROPOSITION 7.1.** *If there exists a primal feasible point to (7.1) with the same pattern as  $A'$ , then the normalized HOTS algorithm converges with a linear rate of convergence.*

*Proof.* In the proof of Theorem 4.6 we did not use the fact that the adjacency matrix  $A$  is a 0-1 matrix, only that its elements are nonnegative. Hence, the convergence proof of the effective HOTS algorithm directly applies to the normalized HOTS algorithm.  $\square$

We shall see in Table 7.1 that the convergence properties of this algorithm seem to be better than those of the classical HOTS algorithm. Nevertheless, these experimental results are still to be validated by other theoretical and numerical studies.

**Acknowledgment.** I gratefully thank Stéphane Gaubert who advised me to apply nonlinear Perron-Frobenius theory to tackle the convergence of HOTS algorithm.

REFERENCES

- [1] MARIANNE AKIAN, STÉPHANE GAUBERT, AND ROGER NUSSBAUM, *Uniqueness of the fixed point of nonexpansive semidifferentiable maps*, arxiv:1201.1536, (2012).
- [2] ABRAHAM BERMAN AND ROBERT J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, no. 9 in Classics in Applied Mathematics, Cambridge University Press, 1994.
- [3] PAOLO BOLDI, BRUNO CODENOTTI, MASSIMO SANTINI, AND SEBASTIANO VIGNA, *UbiCrawler: A scalable fully distributed web crawler*, Software: Practice & Experience, 34 (2004), pp. 711–726. Webgraph available: <http://law.dsi.unimi.it>.
- [4] SERGUEY BRIN AND LARRY PAGE, *The anatomy of a large-scale hypertextual web search engine*, Computer Networks and ISDN Systems, 30 (1998), pp. 107–117. Proc. 17th International World Wide Web Conference.
- [5] P. J. BUSHELL, *Hilbert's metric and positive contraction mappings in a banach space*, Archive for Rational Mechanics and Analysis, 52 (1973), pp. 330–338.

- [6] YAIR CENSOR AND STAVROS A. ZENIOS, *Interval-constrained matrix balancing*, Linear Algebra and its Applications, 150 (1991), pp. 393–421.
- [7] MICHAEL G. CRANDALL AND LUC TARTAR, *Some relations between nonexpansive and order preserving mappings*, Proceedings of the American Mathematical Society, 78 (1980), pp. 385–390.
- [8] B. CURTIS EAVES, ALAN J. HOFFMAN, URIEL G. ROTHBLUM, AND HANS SCHNEIDER, *Line-sum-symmetric scalings of square nonnegative matrices*, in Mathematical Programming Essays in Honor of George B. Dantzig Part II, vol. 25 of Mathematical Programming Studies, Springer Berlin Heidelberg, 1985, pp. 124–141.
- [9] OLIVIER FERCOQ, *Perron vector optimization applied to search engines*, ArXiv e-print:1111.2234, (2011).
- [10] SHMUEL FRIEDLAND, STÉPHANE GAUBERT, AND LIXING HAN, *Perron-frobenius theorem for non-negative multilinear forms and extensions*, Linear Algebra and its Applications, (2011). doi:10.1016/j.laa.2011.02.042.
- [11] STÉPHANE GAUBERT AND JEREMY GUNAWARDENA, *The perron-frobenius theorem for homogeneous monotone functions*, Transactions of the AMS, (2004), pp. 4931–4950.
- [12] ANJELA YURYEVNA GOVAN, *Ranking Theory with Application to Popular Sports*, PhD thesis, North Carolina State University, 2008.
- [13] DARALD J. HARTFIEL, *Concerning diagonal similarity of irreducible matrices*, Proceedings of the American Mathematical Society, 30 (1971), pp. 413–425.
- [14] CHARLES R. JOHNSON, JOEL PITKIN, AND DAVID P. STANFORD, *Line-sum symmetry via the DomEig algorithm*, Computational Optimization and Applications, 17 (2000), pp. 5–10.
- [15] CHARLES R. JOHNSON, DAVID P. STANFORD, D. DALE OLESKY, AND P. VAN DEN DRIESCHE, *Dominant eigenvalues under trace-preserving diagonal perturbations*, Linear Algebra and its Applications, 212–213 (1994), pp. 415–435.
- [16] JAMES P. KEENER, *The perron-frobenius theorem and the ranking of football teams*, SIAM Review, 35 (1993), pp. 80–93.
- [17] JON KLEINBERG, *Authoritative sources in a hyperlinked environment*, Journal of the ACM, 46 (1999), pp. 604–632.
- [18] PHILIP A. KNIGHT, *The sinkhorn-knopp algorithm: Convergence and applications*, SIAM J. on Matrix Analysis and Applications, 30 (2008), pp. 261–275.
- [19] PAUL KNOPP AND RICHARD SINKHORN, *Concerning nonnegative matrices and doubly stochastic matrices*, Pacific J. Math., 21 (1967), pp. 343–348.
- [20] AMY N. LANGVILLE AND CARL D. MEYER, *A survey of eigenvector methods for web information retrieval*, SIAM Review, 47 (2005), pp. 135–161.
- [21] AMY N. LANGVILLE AND CARL D. MEYER, *Google’s PageRank and beyond: the science of search engine rankings*, Princeton University Press, 2006.
- [22] RONNY LEMPEL AND SHLOMO MORAN, *The stochastic approach for link-structure analysis (SALSA) and the TKC effect*, Computer Networks, 33 (2000), pp. 387–401.
- [23] ZHI QUAN LUO AND PAUL TSENG, *On the convergence of the coordinate descent method for convex differentiable minimization*, Journal of Optimization Theory and Applications, 72 (1992), pp. 7–35.
- [24] ROGER D. NUSSBAUM, *Hilbert’s projective metric and iterated nonlinear maps*, in Memoirs of the American Mathematical Society, vol. 75 (391), AMS, 1988.
- [25] ALEXANDER OSTROWSKI, *über normen von matrizen*, Mathematische Zeitschrift, 63 (1955), pp. 2–18.
- [26] ACADEMIC WEB LINK DATABASE PROJECT, *New Zealand university web sites*. <http://cybermetrics.wlv.ac.uk/database/index.html>, January 2006.
- [27] MICHAEL H. SCHNEIDER, *Matrix scaling, entropy minimization, and conjugate duality (i): existence conditions*, Linear Algebra and its Applications, 114–115 (1989), pp. 785–813. Special Issue Dedicated to Alan J. Hoffman.
- [28] ———, *Matrix scaling, entropy minimization and conjugate duality (ii): The dual problem*, Math. Prog., 48 (1990), pp. 103–124.
- [29] MICHAEL H. SCHNEIDER AND STAVROS A. ZENIOS, *A comparative study of algorithms for matrix balancing*, Operations Research, 38 (1990), pp. 439–455.
- [30] WARREN D. SMITH, *Sinkhorn ratings, and new strongly polynomial time algorithms for sinkhorn balancing, perron eigenvectors, and markov chains*. Available: <http://rangevoting.org/WarrenSmithPages/homepage/works.html>, 2005.
- [31] JOHN A. TOMLIN, *A new paradigm for ranking pages on the world wide web*, in Proc. 12th inter. conf. on World Wide Web, WWW ’03, New York, 2003, ACM, pp. 350–355.
- [32] STAVROS A. ZENIOS AND YAIR CENSOR, *Massively parallel row-action algorithms for some nonlinear transportation problems*, SIAM J. Optim, 1 (1991), pp. 373–400.